verlihub Manual

Welcome to Verlihub-0.9.8d. This manual has been modified for Verlihub-0.9.8d but should degrade well with previous versions.

Please rememeber this is a reference manual specifically aimed at explaining the operations and functions of Verlihub as a whole and is not a distribution specific guide. For help with specific distributions you should visit the verlihub forums locatred here.

Further references and examples can be obtained from the Verlihub wiki which can be found here.

What is verlihub?

VerliHub is a Direct Connect protocol server (Hub). It is primarily developed for linux, and is written in C++. It uses very low amounts of cpu-ram-bandwidth relative to other Hub softwares, and offers many useful features. Verlihub accepts numerous plugins and uses a MySQL database to store settings, user, and some runtime information.

Table of contents

---

1. Getting Started

First of all, you will need the source for verlihub from sourceforge. Download the release of your choice and save it in /root (root home folder). Throughout the course of this manual, it will be assumed you are logged in as root during the compiling and installation phase and the source is located in the root home folder. (**Note:** root access is not generally needed during compiling but is preferred in this guide due to the multitude of different options provided per distribution. If you know what you are doing and want to use a different username, make sure you correct all the path references from here on.)

Use the following command to extract the files:

```
tar zxvf verlihub-0.9.8d_RC1.tar.gz
```

This will extract the source into it's own folder. (In the example above, it will be in /root/verlihub-0.9.8d).

2. Installation

There are some requirements that need to be fulfilled before you attempt to compile Verlihub. **Each linux distribution has it's own type of package management system, it is up to you to research it if you decide to use a package management system to install the dependancies. The following dependancy guide will be given on the assumption that we are using Gentoo Linux.**

- gcc version >= 3.2 (gcc --version will tell you the version number)
- mysql (3.23 or higher) **Slackware users only: use 4.0.20 or higher**
- Perl-compatible regular expression library (pcre)
- GeoIP

- ***Optional:*** A remote way to administer mysql. (This guide is not dependent on it, and it **IS NOT** a requirement. For those that use mysql, this can make life a lot easier.)

The package distribution needs to be updated prior to installing any packages. For gentoo, use:

```
emerge sync
```

Then proceed to the next section.

2.1 Installing geoIP

Google around to locate the source, or use rpmfind.net if your distribution supports it. For gentoo users, installing geoip using portage is simple; use:

```
emerge geoip
```

This will download, compile and install it for you.

2.2 Installing Perl-compatible regular expression library (pcre)

This is what is required: libpcre0-devel or libpcre1-devel. Again, Google around to locate the source, or use rpmfind.net if your distribution supports it.

For gentoo users, use:

```
emerge libpcre
```

This will download, compile and install it for you.

2.3 Installing mysql

Once again, Google around to locate the source, or use rpmfind.net if your distribution supports it.

Yet again, you can install it using the package management system:

```
emerge mysql
```

This will download, compile and install it for you. After mysql installs, there still are three things you need to do. Firstly, mysql doesn't come with any database files installed. For gentoo users, use:

```
mysql-install-db
```

to install the database files.

Second, mysql should start at boot up. Use the following gentoo distribution command (other

distributions will have a command similar to it):

```
rc-update add mysql default
```

and this will automagically add mysql to your startup scripts.

Lastly, you need to start mysql. You can do this by rebooting or use:

```
/etc/init.d/mysql start
```

to start it.

You should remove the two anonymous user accounts from mysql. Use this to accomplish it:

```
mysql -u root
mysql> DELETE FROM mysql.user WHERE User = '';
mysql> FLUSH PRIVILEGES;
```

Lastly, you **have** to set a root password for mysql, and remove the anonymous accounts installed by default. Reset the root password by using:

```
mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');
```

Replace the password between the two single quotes with a password of your choice. **You will need this password later on!** For the second password statement, replace 'host_name' with the IP address of the computer mysql is running on.

Now mysql is ready to go. Keep in mind that mysql does not necessarily need to run on the same machine as the hub software but it IS required to be installed for compiling the hubsoft. It can be located (preferably) on the same LAN for latency reasons. Instructions to set this up however are outside the scope of this document.

2.4 Optional: Installing phpmyadmin

This step is optional. If you do not need phpmyadmin then skip this step and go to *2.5 Preparing and compiling the source*.

phpmyadmin is a set of scripts written in PHP that allows you to use apache to remotely administer your mysql database. This can prove to be quite handy should you need to change some values in the verlihub database directly. Of course, there are two prerequisites to running phpmyadmin: you need apache and php installed. While installing and configuring apache, php and phpmyadmin are outside the scope of this document, gentoo's portage system makes it easy to install and configure, so these three steps will be shown here.

Step 1

First, you need to make sure /etc/hosts is configured correctly. The hostname of the computer must reside next to the 127.0.0.1 listing. To get your hostname, do:

```
cat /etc/hostname
```

This will show you your hostname. Now, open /etc/hosts in your favorite editor and make sure your localhost (127.0.0.1) line looks similar to this:

```
127.0.0.1 localhost myhostnamefromabove
```

DO NOT remove localhost from this line! Simply go to the end of the line, press the spacebar and enter in your hostname.

Step 2

Second, get apache and mod_php installed (we will assume php is used as a module here.) Use the following command to install them:

```
emerge apache mod_php
```

This will install both packages. Now all that is needed is some quick configuration. Open **/etc/conf.d/apache2** in your favorite editor and change:

```
#APACHE2_OPTS="-D SSL"
```

to

```
APACHE2_OPTS="-D PHP4"
```

Ensure you remove the '#' (pound character) from this line. Now open **/etc/apache/conf/apache2.conf** in your favorite editor and make two changes. Change:

```
#ServerName=localhost
```

to

```
ServerName=localhost
```

then change the *ErrorLog* line (this should only be a few lines below the **ServerName** line above) to:

```
ErrorLog /var/log/apache2/error_log
```

Make sure the directory /var/log/apache2 exists. If not, create it.

To start the webserver use:

```
/etc/init.d/apache2 start
```

To stop it use:

```
/etc/init.d/apache2 stop
```

If you want it to be loaded at startup, use:

```
rc-update add apache2 default
```

Step 3

The final step: install phpmyadmin. Use:

```
emerge phpmyadmin
```

to accomplish this. **NOTE:** Watch the output of emerge! You need to run some commands in order to have phpmyadmin work! It involves importing some SQL commands into mysql.

After all this is done browse to http://localhost/phpmyadmin if you are running a web browser on the PC that verlihub is installed on, or http://<myhostname>/phpmyadmin if from another PC. **NOTE:** use the root username/password to login!

2.5 Preparing and compiling the source

At this point in this guide, the commands are not gentoo-distribution specific; they should work across different platforms unless specifically noted.

Now that the prerequisites are installed, there should be no problems building the source. To build the source use:

```
cd /root/verlihub-0.9.8d
./configure
make
```

This could take several minutes to compile depending on the speed of your processor. After it's done, see Installing the fresh build.

2.6 Installing the fresh build

If the compile has completed successfully, then install it using:

```
make install
```

Some environment variables may have changed. You may need to restart your computer. If the compile did not complete successfully please refer to the Verlihub Forums.

Gentoo users can use this command and continue with no reboot:

```
env-update
```

DO NOT START THE HUB YET! Proceed to the next section for some initial configuration of the

hub before the initial start-up.

3. Starting up for the first time

In order to start the hub we must first run a configuration script. If the above steps completed successfully then you should be able to run the scripts from anywhere.

```
cd /root
vh_install
--------------------------------
Your name ? (root) press enter

Hello root,
let's start with configuration of database access..

--------------------------------
mysql database for verlihub will be called? (verlihub) press enter to
accept the default or change according to your wishes
mysql user to access verlihub gonna be? (verlihub) press enter to
accept the default or change according to your wishes
password to access verlihub be? (1110079846) press enter to accept the
default or change according to your wishes
mysql server will run where? (localhost) press enter to accept the
default or change according to your wishes
--------------------------------
user = verlihub
password = 1110079846
host = localhost
database = verlihub
--------------------------------
This database account cannot be accessed
--------------------------------
Is this info correct ? (Y/N)y
Do you want to create database now? (Y/N)y
--------------------------------
Ok preparing mysql..
You probably need administrator access to mysql database
mysql administrator username? (root)press enter to accept the default
or change according to your wishes

--------------------------------

You'll be now promted by mysql client for password of root@localhost
Enter password:enter mysql root password or the password for the mysql
administrator
--------------------------------
This database account exists
```

```
--------------------------------

--------------------------------

root, you need to choose a place for the configuration files
--------------------------------
The order of folder that verlihub is looking for is following:

1 - variable $VERLIHUB_CFG - -inexisting-
2 - ./.verlihub - /root/verlihub-0.9.8c/scripts/.verlihub -inexisting-
3 - /root/.verlihub - /root/.verlihub -inexisting-
4 - /usr/local/etc/verlihub - /usr/local/etc/verlihub -inexisting-
5 - /etc/verlihub - /etc/verlihub -inexisting-
if two or more of these exist, lower number has priority
--------------------------------

--------------------------------
what is will be the configuration folder ? (/etc/verlihub)press enter
to accept the default or change according to your wishes
The config folder /root/.verlihub does not exist and would be created
--------------------------------
Do you want to continue with these settings (if not then select
another folder) ? (Y/N)y
--------------------------------
Written: /root/.verlihub/dbconfig

--------------------------------

Wait a few seconds..
./vh_install: line 63: <PID> Killed $bindir/verlihub 1 >&/dev/null
--------------------------------
root, now I will ask you few more questions about your future hub, if
you permit..

--------------------------------
Try to not put many special characters, you'll be able to put some laer

Give me your DC hub master nickname.. ([SU]root)think of a master
nickname and enter it here
Choose your password.. (1110079846) enter a password for the master
nickname
Which will be default ONE hub port number? (4111)[Choose a port; <
1024 verlihub needs to run as root. Default: 4111, DC uses 411.]
What will be your hub hostname? (<computername>)enter the url of your
hub
Give me the name of your hub (hub of root) enter the name of your hub
--------------------------------
Hub: '<hubname entered above>'
```

```
url: 'dchub://<hostname entered above>:<port entered above>'
Master user: '<master username entered above>'
Master's password: '<master password entered above>'
------------------------------
Is this info correct ? (Y/N)y
FYI: settings are going to be created or updated
/etc/verlihub
/etc/verlihub
/etc/verlihub
will invoke the command
class is 10
nick is <master username entered above>
password is <master password entered above>
done
```

3.1 What port should verlihub run on?

The installer script asks for a port to use. This section is here only to list the two common choices, and will show how to change the port should you decide to use a different one. When Verlihub is installed, it will use **port 4111** by default. This allows the verlihub daemon to run as a non-root user, which can be desirable for security reasons. Most hub installs should be fine with this. If verlihub is behind a firewall, the firewall may be able to route incoming requests from port 411 to the verlihub machine listening on port 4111. However, DC *clients* will always use **port 411** if a port is not specified with the hub address. When specifying a hub address with a port, the format is *<hub address>:<port>*. The only concern with this is that the verlihub daemon has to run as root to bind to this port.

Verlihub should be ready to go now; if you want to change the port, start verlihub directly:

```
/usr/local/bin/verlihub
```

Then issue the following command from inside the hub to change the port:

```
!set listen_port <port>
```

Restart Verlihub to use the new port.

4. Running Verlihub

Now that the Hub has been set up, now you need to decide how it starts. In this guide so far, verlihub has been invoked manually. It can also be run as a daemon.

4.1 Running normally

This is how it has been invoked so far. You can log on as a normal user and start the hub manually each time by using:

```
/usr/local/bin/verlihub
```

When running the hub this way, all you need to do is press CTRL+C to stop the hub software, or if you are logged in as the Master user type !quit in the chat window. **Please note:** When you run the hub as a non-root user you CANNOT set the hub to listen on ports below 1024!

4.2 Running as a daemon

When verlihub is configured to run as a daemon, it is generally loaded in the startup scripts. You don't need to be there manually to login and start up the hub. The problem is that different distributions have different ways of writing scripts and adding them to the startup sequence. You should really check out the forums (link is at the top of this page) for possible help with certain distributions.

Verlihub comes with a script to launch as a daemon, but it may not work in all instances (see Starting and Stopping verlihub as a daemon for how try try to use the provided script.) The following is **gentoo-distribution specific**. This is a script that can be used with gentoo to start and stop verlihub. First, copy this into **/etc/conf.d/verlihub**:

```
# Copyright 1999-2005 Gentoo Foundation
# Distributed under the terms of the GNU General Public License v2
# $Header: /cvsroot/verlihub/verlihub/docs/html_manual/verlihub_manual.
html,v 1.3 2007/03/03 15:21:52 chaosuk Exp $

# Verlihub configuration

# The directory where the verlihub config files are
CONFDIR=/etc/verlihub

# The location of the verlihub binary
VERLIBIN=/usr/local/bin/verlihub

# The process file
VERLIPIDFILE=/var/run/verlihub.pid

# Required files for verlihub to start
REQFILES="dbconfig"

# Optional files for verlihub
OPTFILES="motd faq rules help_admin help_master help_op help_usr
help_reg"
```

Then save this into **/etc/init.d/verlihub**:

```
#!/sbin/runscript
```

```
# Copyright 1999-2005 Gentoo Foundation
# Distributed under the terms of the GNU General Public License v2

# Verlihub boot script for gentoo distributions

depend() {
        need net
        need mysql
}

start() {
        ebegin "Starting verlihub"

        # make sure the configuration directory exists
        if [ ! -d ${CONFDIR} ]; then
                eerror FATAL ERROR: missing configuration directory
${CONFDIR}
                return 1
        fi;

        # check for the required files
        for f in ${REQFILES}; do
                file=${CONFDIR}/$f

                if [ ! -e $file ]; then
                        eerror "FATAL ERROR: missing required file
$file"
                        return 1
                fi;
        done;

        # check for the optional files
        for f in ${OPTFILES}; do
                file=${CONFDIR}/$f
                if [ ! -e $file ]; then
                        eerror "WARNING: missing file $file, some
functionality will be disabled"
                fi;
        done;

        # start the hub
        start-stop-daemon --start --quiet --background --make-pidfile
--pidfile ${VERLIPIDFILE} --exec ${VERLIBIN}

        eend $?
}
```

```
stop () {
        ebegin "Stopping verlihub"

        # stop verlihub
        start-stop-daemon --stop --pidfile ${VERLIPIDFILE}

        eend $?
}
```

After you save that script, you need to make it executable. Use:

```
chmod +x /etc/init.d/verlihub
```

to do this. Now we need to add the new initscript to the startup scripts. Use:

```
rc-update add verlihub default
```

to do this. See the next section for how to use the gentoo script.

NOTE: This initscript is intended to be run as root (so port 411 can be used.) If you want the initscript to run as a user, first create the user verlihub on your machine, then change this line in **/etc/init.d/verlihub**:

```
start-stop-daemon --start --quiet --background --make-pidfile --
pidfile ${VERLIPIDFILE} --exec ${VERLIBIN}
```

to

```
start-stop-daemon --start --quiet --background --make-pidfile --
pidfile ${VERLIPIDFILE} --chuid verlihub --exec ${VERLIBIN}
```

4.2.1 Starting and Stopping verlihub as a daemon

For other scripts see Helper scripts for more information. These scripts are in CVS and versions dated > August 26, 2004.

In the /scripts folder there is a runhub command. Be warned that this script **may not work** in all instances! To start it use:

```
vh_runhub
```

To try to stop it with the runhub script try:

```
vh_runhub -s
```

The following is **gentoo-distribution specific**, and you need to set up the script in the previous step for this to work.

To start verlihub:

```
/etc/init.d/verlihub start
```

To stop verlihub:

```
/etc/init.d/verlihub stop
```

To restart verlihub:

```
/etc/init.d/verlihub restart
```

4.3 Helper scripts

There are helper scripts in the CVS version of verlihub (and/or versions later than August 26, 2004.) They have been listed here in a table with a brief description of what they are capable of:

Script	Description
vh_getcfg	Run this script and see which configuration folder verlihub is most likely to use
vh_runhub	start, stop, restart verlihub as a daemon; creates log and err files; displays missing files
vh_getdb	Finds the dbconfig files and parses it, then allows you to do multiple things with that information such as: connect; query; gethost; getdata; getuser; getpass
vh_regnick	Creates a registered user
vh_trigger	Creates a file_trigger

5. Configuring your Hub

All of the settings are stored in the database. All administration is done from within the hub, by a person logged on as the Master user.

To get the current configuration settings type:

```
!getconfig
```

To change a setting to a new value type:

```
!set <variable> <value>
```

Once changes are made, you will need to refresh the hub. (Some changes will require you to stop and start the hub again, however.) To do this type:

```
!reload
```

If your changes don't apply, restart the hub manually.

5.1 Basic Configuration

There are a few components to the basic configuration of your hub. There is some settings that need to be changed in the database, as well as some files need to be added (such as a Message of the Day, FAQ, amongst other things...) to the configuration directory.

5.1.1 Hub Variables

In the following table is a list of the "basic" things that should be set in your hub.

Configured with install script	
Variable	**Explanation**
listen_port	The port the hub listens on. Other hubs use port 411, but you need to run verlihub as root to do that. Default: 4111.
hub_host	What people will use to connect to your hub. (An example: myhub.no-ip.com)
hub_name	The name of your hub.
Not configured - you may need to configure these	
Variable	**Explanation**
listen_ip	If you have more than one network card in your PC, enter the IP address to listen on. If you don't have more than one network card, no change is necessary. Default: (empty)
hub_desc	A description of your hub.
hub_category	An upcoming extension to the hublist protocol this variable allows you to specify what type of hub you have i.e. movies
hub_owner	Owner that runs the hub.
min_nick	Minimum nickname length. Default: 3.
max_nick	Maximum nickname length. Default: 32.
max_users	Maximum number of users allowed in your hub. Default: 6000
min_share	Minimum amount that people must share in your hub in megabytes. People that share less will not be allowed to join. Default: 2 GB (2048).
tban_kick	Length of time people must wait after being kicked to rejoin. In seconds. Default: 300 seconds. (5 minutes)
send_user_info	When people join, the hub will send them information about themselves. Set to 0 (zero) to disable, or 1 to enable. Default: 1.
send_user_ip	When people join, the hub will send them the IP address they are reporting. Set to 0 (zero) to disable, or 1 to enable. Default: 0.
hub_security	This is the name of the Hub Security bot.
hub_security_desc	This is the description of the Hub Security bot.

opchat_name	This is the name of the OpChat bot.
opchat_desc	This is the description of the OpChat bot.

For example, to set the hub name to 'My Hub' use this command:

```
!set hub_name My Hub
```

5.1.2 Hub Topic

To set a topic in the hub (this is usually displayed in the title bar of the client) use:

```
!topic This is a useless topic
```

This command will change the topic to "This is a useless topic". By default only administrators can set the hub topic. If you wish to allow classes lower then admin (level 5) to set the topic, you can specify this in your settings.

5.1.3 Message of the Day

The message of the day will be shown to every member joining the hub. All that needs to be done is a text file needs to be saved as 'motd' in the configuration directory (in this guide /etc/verlihub/motd) and verlihub will automatically use it.

Verlihub allows you to show extra motd's based on user class. Here is a file list you can use:

File	Description
motd	All users receive this motd.
motd_reg	Registered users receive this **in addition to** motd.
motd_vip	VIP Users receive this **in addition to** motd_reg and the normal motd.
motd_op	OPs receive this **in addition to** motd_vip, motd_reg and the normal motd.
motd_cheef	Cheef OPs receive this **in addition to** motd_op, motd_vip, motd_reg and the normal motd.
motd_admin	Admin OPs receive this **in addition to** motd_cheef, motd_op, motd_vip, motd_reg and the normal motd.
motd_master	Master users see all motd messages.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**. See here for more information.

5.1.4 FAQ

A FAQ (Frequently Asked Questions) is available to users using +faq in the hub chat window. Again, all that needs to be done is a text file needs to be saved as 'faq' in the configuration directory (in this guide / etc/verlihub/faq) and verlihub will automatically use it.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**. See here for more information.

5.1.5 Hub Rules

A rules trigger is also automatically provided. When people type +rules in the hub chat window, the contents of 'rules' in the configuration directory (in this guide /etc/verlihub/rules) are sent to the user.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**. See here for more information.

5.1.6 Hub Help

In order to have help available to users, you need to create 7 different help files. At the time of this writing, starter help files are available in the forums; go there and to a search. As this thread may change, it will not be hotlinked. The file available there has also been mirrored here. **NOTE:** This file also contains templates for *rules*, *motd*, and *FAQ*. Be careful extracting if you have existing copies of these!

Before you do this, it may help to understand how verlihub deals with help files. There are several "classes" of users that can exist in verlihub. Each one has a help file associated with it. The help will be sent by class, for example, a regular user will only get class 0 help when they use +help. A regular operator will get class 3, 2, 1, and 0 help sent to them when they use +help. Because of this, it is important to create all the help files, as certain commands are only available to certain users. The files need to be placed in the configuration directory (in this guide it's /etc/verlihub). Here is a table outlining the files needed for the help system to work:

Class	Tag	Help file
0	Regular users	/etc/verlihub/help_usr
1	Registered users	/etc/verlihub/help_reg
2	VIP users	/etc/verlihub/help_vip
3	Operator user	/etc/verlihub/help_op
4	Cheef user	/etc/verlihub/help_cheef
5	Admin user	/etc/verlihub/help_admin
10	Master user	/etc/verlihub/help_master

As for a guide for what to put in these files, refer to the Hub Command list for examples of what to put in these files.

Note: If you don't use CRLF encoding in your text file, the Windows clients will display the whole file **on one line**. See here for more information.

5.1.7 FAQ/Rules/Help/MOTD/File Trigger Encoding

All of the text files verlihub uses should be DOS-encoded. If you use UNIX-encoding, all the windows clients will interpret the text files incorrectly and display the entire file on one line, which is probably not desired (of course, if you only have 1-line FAQ/motd/help/rules it doesn't matter.)

There are tools available to convert file types back and forth. **unix2dos** and **dos2unix** will convert back and forth between the two encodings, however these tools may not be present on your distribution. **Gentoo users** can use this portage command to get the two tools:

```
emerge dos2unix unix2dos
```

This will install the two tools needed. These tools are also widely available for all other distros.

If you use the editor **vim**, it can save and manipulate DOS-encoded files directly. Start vim, and issue this command before saving files:

```
:set fileencoding=dos
```

5.2 Advanced Configuration

The items here are not necessary to configure for the hub to function. It is a good idea to be aware of some of these configurable options, however. All of these are configured as the Master user logged into the hub.

5.2.1 Listening on more than one port

Verlihub has the ability to listen on more than one port for incoming clients, if you need to do this. For example, to listen on 1411 and some others **(NOTE the space between ports when using more then one)** use this command:

```
!set extra_listen_ports 1411 8888 666
```

5.2.2 File Triggers

Verlihub has an advanced file trigger feature. You can add different types of triggers; for example, if you prefix the command with '!' only ops can use the trigger; if you prefix with '+' then users can use the trigger. These are the commands available to use with the file triggers:

Command	Description
!lsttrigger	Lists available triggers.
!addtrigger <trigger> [-d <definition>] [-h <description help>] [-f <flags (see below>] [-n <send as nick>] [-c <min_class>] [-C <max_class>]	Adds a trigger. See below this table for more information regarding the parameters.
!modtrigger <trigger> [-d <definition>] [-h <description help>] [-f <flags (see below>] [-n <send as nick>] [-c <min_class>] [-C <max_class>]	Changes a trigger. See below this table for more information regarding the parameters.
!deltrigger <trigger>	Removes a trigger.

Explanations of command parameters:

<table>
<tr><th>Parameter</th><th>Explanation</th></tr>
<tr><td><trigger></td><td>The name you want the trigger to be called.</td></tr>
<tr><td>-d <definition></td><td>The definition of the trigger. This can be pointing to a file, or it can be the actual trigger contents, if the correct flags are specified. If you use this for the actual contents you will have to enclose it in quotes. An example: -d "This is the trigger contents."

NOTE: When using the definition as the actual trigger contents, you can not have quotations in the file! You will need to edit the database directly in this case.

You can use %[CFG] and verlihub's current configuration path will be used.

These variables need flag bit 32 set:
<table>
<tr><th>Variable</th><th>Description</th></tr>
<tr><td>%[CC]</td><td>User's country code</td></tr>
<tr><td>%[IP]</td><td>User's IP address</td></tr>
<tr><td>%[HOST]</td><td>User's hostmask (dns_lookup must be set to 1 for this to work)</td></tr>
<tr><td>%[USERS]</td><td>The number of users in the userlist</td></tr>
<tr><td>%[UPTIME]</td><td>Hub uptime</td></tr>
<tr><td>%[VERSION]</td><td>Hub version</td></tr>
<tr><td>%[VERSION_DATE]</td><td>Hub's release date</td></tr>
<tr><td>%[HUBNAME]</td><td>The hub name</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>%[NICK]</td><td>User's nickname</td></tr>
<tr><td>%[SHARE]</td><td>User's share size</td></tr>
<tr><td>%[TOTAL_SHARE]</td><td>Total amount shared in hub</td></tr>
</table>
</td></tr>
<tr><td>-h <description help></td><td>This is not used by Verlihub, it is here in case you need to enter a note with the trigger to remember what it is for.</td></tr>
<tr><td>-f <flags></td><td>This is a flag bitset. These will govern how Verlihub will use the trigger and/or how it displays it to the user. This is a table with the available values:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Send to main chat (visible to user only)</td></tr>
<tr><td>1</td><td>Execute command</td></tr>
<tr><td>2</td><td>Message is sent to PM</td></tr>
<tr><td>4</td><td>Automatically trigger when user logs in</td></tr>
<tr><td>8</td><td>Trigger on +help/!help command</td></tr>
<tr><td>16</td><td>The definition is the text</td></tr>
<tr><td>32</td><td>Allow replacing of variables</td></tr>
<tr><td>64</td><td>Message is sent to everyone in the main chat window</td></tr>
</table>
Some examples of usage:
<table>
<tr><th>Verlihub will do this</th><th>Flags to use</th></tr>
<tr><td>Send to user only in main chat</td><td>-f 0</td></tr>
<tr><td>Send message to user on login, and to a PM window</td><td>-f 6</td></tr>
<tr><td>Trigger on +help and send it to PM window</td><td>-f 10</td></tr>
<tr><td>Send message on login and on +help/!help to a PM</td><td>-f 14</td></tr>
<tr><td>Use definition as trigger contents and send to PM window with variables</td><td>-f 50</td></tr>
<tr><td>Send to everyone in main chat window</td><td>-f 64</td></tr>
</table>
</td></tr>
<tr><td>-n <send as nick></td><td>When sending to the main chat window, the trigger will appear to send from this nickname.</td></tr>
<tr><td>-c <min_class></td><td>This is the minimum user class allowed to use this trigger.</td></tr>
<tr><td>-C <max_class></td><td>This is the maximum user class allowed to use this trigger.</td></tr>
</table>

For example, to add a trigger called '+hublisting' from a file and have it go to a PM window when invoked:

```
!addtrigger +hublisting -d "/path/to/file" -f 2
```

TIP: Did you know that the +motd/+help/+faq/+rules behaviour can be changed using the above flags? For example, if you want to send the FAQ to a PM, use this command:

```
!modtrigger +faq -f 2
```

NOTE: The *!ftedit* command is no longer supported.

5.2.3 Flood protection

Verlihub has diverse parameters to protect the hub and the mainchat from flooding. There must be some compromise between heavy restrictions and the possibility to chat. In the latest version of Verlihub a default protection feature has been implemented which prevents ANY non regged user from sending the same raw information twice in succession weather its chat, a pm message, a search string, a TTH search string, etc etc. This CANNOT be switched off.

There are several hub variables that can help prevent mass flooding and abuse of mainchat. *max_chat_msg* can determine the maximum length of a message allowed to be sent to mainchat (default is 256 characters long). *max_chat_lines* can determine the maximum number of lines that can be sent to mainchat (default is 5). Whereas *mainchat_class* can actually lock the mainchat *on the fly* to a specified class of user. As an example, to set a maximum length of 512 characters and 10 lines with only registered users allowed to chat:

```
!set max_chat_msg 512
!set max_chat_lines 10
!set mainchat_class 1
```

If you have active bots that are frequently kicking naughty users, you may want to get rid of the kicking messages that constantly spam your mainchat area.. You can use the variable *hide_all_kicks* to do this; 1 hides all kick messages, 0 shows messages. Use this to hide all messages:

```
!set hide_all_kicks 1
```

Another possibility for muting kick messages is using commands **!hidekick <nick>** and **!unhidekick <nick>** which will kick the one user and not display anything.

Verlihub does not prevent users saying rude words. However, the forbidden words plugin will allow you to prevent this, or you may consider using the replacer plugin which will actively scan every mainchat message and replace predetermined words with specified ones. Please visit the project homepage for more information on plugins.

5.2.4 Hub Redirections

You can set up to 10 hubs where verlihub will try to redirect every disconnected user. The choice is made randomly among all ten of them although the first slot is likely to be chosen more oftern then the others.

It is IMPORTANT that you set a redirect host for your hub although not imperitive. Please be warned if you have a busy hub and wish to disable redirects by setting the first redirect slot to your own hub address, keep in mind that the users you are trying to redirect have ALREADY BEEN REJECTED by your hub per your minimum requirements for entry. Leaving these users in a connection loop will not help your stability or connection pool. The example below demonstrates how to disable redirects.

```
!set redir_host0 my-hub.no-ip.com (replacing my-hub.no-ip.com with
your hub host address)
!set redir_host_max 0
```

As mentioned above there are 10 possible slots where an address can be set for random redirects. Please note: In order to enable these extra redirect address slots you must set the *redir_host_max* variable accordingly. Example: I have filled slots 0 through to 3 so I would set the *redir_host_max 3* and NOT 4 as this would enable an empty slot leaving open the possibility of dead user connections. (something you dont want)

```
!set redir_host0 somehub.org
!set redir_host1 someotherhub.org
!set redir_host2 yetanotherhub.org
!set redir_host3 yetanotherhub1.org
!set redir_host4 yetanotherhub2.org
!set redir_host5 yetanotherhub3.org
!set redir_host6 yetanotherhub4.org
!set redir_host7 yetanotherhub5.org
!set redir_host8 yetanotherhub6.org
!set redir_host9 yetanotherhub7.org
```

5.2.5 Customizing Error Messages

Verlihub allows you to customize the following error messages:

- User is banned message (msg_banned)
- Hub full message (msg_hub_full)
- Invalid nickname prefix (msg_nick_prefix)
- Client version too new (msg_downgrade)
- Client version too old (msg_upgrade)

Here is an example of how to set the messages:

```
!set msg_banned You are banned from this hub.
```

```
!set msg_hub_full This hub is full, try again later.
!set msg_nick_prefix You do not have the desired nickname prefix in
order to join this hub.
!set msg_downgrade You are using too new of a client. This hub does
not support untested clients at this time.
!set msg_upgrade You are using an outdated client. Upgrade your client
and rejoin.
```

5.2.6 Customizing kick/ban Messages

There is a variable *msg_replace_ban* that can be set to replace the '_ban_*' parts of the kick message so they are not displayed to the hub. By default, this variable is an empty-length string ('') so that part of the message is removed. If you want, you can set this to replace the '_ban_*' part of the message with whatever you want. For example, if you banned someone with this command and msg_replace_ban is set to 'Replace':

```
!kick <nick> You have been a bad user. _ban_2d
```

The users in the hub would see this kick message:

```
[<time>] <OPUser> is kicking user <nick> because: You have been a bad
user. Replace
```

To change the replace text with something else, use:

```
!set msg_replace_ban <text>
```

5.2.7 Customizing Welcome Messages

Verlihub allows you to greet members as they join. Newly registered users will be greeting with a message to change their password, and if the main chat window is disabled in the hub, they will see the command for how to turn the chat window on and off.

To change the two default messages (password and chat) use this:

```
!set msg_chat_onoff "Use !chaton to display messages in the chat
window."
!set msg_change_pwd "You have 5 minutes to change your password."
```

As far as welcoming users, you can welcome users specially by using:

```
!set msg_welcome_guest Welcome, %[nick] has joined!
!set msg_welcome_reg Registered member %[nick] has joined!
!set msg_welcome_vip VIP member %[nick] has joined!
!set msg_welcome_op OP %[nick] has joined.
!set msg_welcome_cheef OP %[nick] has joined.
!set msg_welcome_admin Admin OP %[nick] has joined.
!set msg_welcome_master Hub owner %[nick] has joined. Everyone hide!
```

NOTE: msg_welcome_guest is for anyone who joins that is not a Class 1 user or up.

Tip: If you want to use the user's nickname in the message, use the variable %[nick] as shown above; use %[CC] for the user's country code.
There is also the ability to set a custom message for a registered user that he/she will see when entering the hub. This will only be shown to that user and not the entire hub. The example below shows how to do this:

```
!regset <username> msg_welcome <your custom message>
```

5.2.8 Advanced Share Limit control

Verlihub allows you to set different share levels for registered users and hub operators. You can also specify an overall maximum allowed share, and a maximum allowed share for registered users (all are in megabytes). For example, to set a minimum share of 2 gigabytes for registered users, 1 gigabyte for VIP's and no share for OPs, use:

```
!set min_share_reg 2048
!set min_share_vip 1024
!set min_share_ops 0
```

To set an overall maximum share of 1000 gigabytes and a maximum share of 500 gigabytes for registered users, 1000 gigabytes for VIP's and 0 gigabytes for OP's (forcing them to NOT share) use:

```
!set max_share 1024000
!set max_share_reg 5176000
!set max_share_vip 1024000
!set max_share_op 0
```

EXTRA INFORMATION:
In this version there is also the ability to hide a registered users share size from the nicklist. Use the below example to achieve this: (1 = hide users share on next login and 0 = show users share on next login.)

```
!regset <username> hide_share 1
```

5.2.9 Restricting Clients

Verlihub also recognizes DC tags. Using these tags, you can restrict users access to your hub.

NOTE: For users of Verlihub older than 0.9.8 the method for restricting clients has changed.

Verlihub now supports a dynamic way to restrict clients based on the type of connection they use. There are four commands that manipulate the tags and the restrictions that each tag imposes. The commands are summarized below:

Command	Explanation
!lstconntype	Shows the current DC tags and their restrictions.
!addconntype <DC Tag> [-S <tag_max_slots>][-s <tag_min_slots>][-l <tag_min_limit>][-ls <tag_min_ls_ratio>]	Adds the DC tags specified and their restrictions. See the explanation of the parameters in the table below.
!modconntype <DC Tag> [-S <tag_max_slots>][-s <tag_min_slots>][-l <tag_min_limit>][-ls <tag_min_ls_ratio>]	Adds the DC tags specified and their restrictions. See the explanation of the parameters in the table below.
!delconntype <DC Tag>	Removes specified DC Tag.

This table explains the parameters used to change the restrictions for each DC tag:

Parameter	Explanation
-S <tag_max_slots>	Maximum slots user can define for this connection type
-s <tag_min_slots>	Minimum slots user can define for this connection type
-l <tag_min_limit>	Minimum upload limit for user with this connection type. If user upload rate is **less** than this value, they are not allowed in the hub.
-ls <tag_min_ls_ratio>	Minimal value for upload limit per slot. If user upload rate per slot is **less** than this value, they are not allowed in hub.

NOTE: A set of these tags have been pre-installed for you to use. Use **!lstconntype** to show them.
REMEMBER: The *default* tag is used when a matching DC Tag is not found in the list!

5.2.9.1 Hub/slot ratios & Maximum hubs allowed in

Most hubs allow you to restrict how many hubs your users are in, and also a hub/slot ratio. Verlihub is no different, for example, if you want to restrict your users to be in 4 hubs maximum, and have 1 slot per hub, use:

```
!set tag_max_hs_ratio 1
!set tag_max_hubs 4
```

5.2.9.2 Minimum/Maximum slots

Verlihub allows you to restrict the number of slots the client is using. You can specify default minimum and maximum values, as well as values that are specific to the type of connection. Below is an example of setting the minimum slots to 1 and the maximum to 50:

```
!modconntype default -s 1 -S 50
!modconntype 28Kbps -s 1 -S 50
!modconntype 33Kbps -s 1 -S 50
!modconntype 56Kbps -s 1 -S 50
!modconntype Modem -s 1 -S 50
!modconntype ISDN -s 1 -S 50
!modconntype Cable -s 1 -S 50
!modconntype DSL -s 1 -S 50
!modconntype Satellite -s 1 -S 50
!modconntype Microwave -s 1 -S 50
!modconntype Wireless -s 1 -S 50
!modconntype LAN(T1) -s 1 -S 50
!modconntype LAN(T3) -s 1 -S 50
```

5.2.9.3 Speed per available slot

Verlihub also allows you to set a minimum speed **per slot** for the client. Users that are below it are not allowed a connection to the hub. You can specify an overall default, as well as connection-specific limits.

Below is an example how to *disable* this limit. If you want to set them, enter values greater than 0:

```
!modconntype default -ls -1
!modconntype 28Kbps -ls -1
!modconntype 33Kbps -ls -1
!modconntype 56Kbps -ls -1
!modconntype Modem -ls -1
!modconntype ISDN -ls -1
!modconntype Cable -ls -1
!modconntype DSL -ls -1
!modconntype Satellite -ls -1
!modconntype Microwave -ls -1
!modconntype Wireless -ls -1
!modconntype LAN(T1) -ls -1
!modconntype LAN(T3) -ls -1
```

Tip: If you do not want to use the limiting, set the values to **-1**.

5.2.9.4 Upload capping

If the client connected is capable of upload capping, you can specify a minimum cap allowed before they are allowed to join in the hub. All that needs to be set is the default cap, however, you can also specify caps depending on the type of connection the user is reporting. These values are in KB/sec.

In the example below, a 25KB/sec default cap is set, and various caps are set depending on the connection:

```
!modconntype default -l 25
!modconntype 28Kbps -l 2
!modconntype 33Kbps -l 3
!modconntype 56Kbps -l 5
!modconntype Modem -l 5
!modconntype ISDN -l 13
!modconntype Cable -l 13
!modconntype DSL -l 13
!modconntype Satellite -l 20
!modconntype Microwave -l 20
!modconntype Wireless -l 20
!modconntype LAN(T1) -l 50
!modconntype LAN(T3) -l 100
```

Tip: If you do not want to use the limiting, set the values to **-1**.

5.2.10 Hublist registering

Verlihub has also a very simple and a bit incomplete hublist registering support. Well there are several variables and one command that make it work. Those that are used are:

- hublist_host - the ip address of the hublist registering host
- hublist_port - usually 2501 which is default
- hub_host
- listen_port
- hub_name
- hub_desc
- hub_category - a forthcoming update of the hublist protocol that will be implemented by www.dchublist.com and others

The command to send the registration off is:

```
!hublist
```

You can use this command once the above variables are set correctly. Do not use it more than once every 20 minutes, as the hublists tend to ban hubs that update too frequently.

5.2.11 Setting up a Registered Users only Hub

Verlihub can restrict users to member-only visitors. For example, use this to not allow unregistered users and 1000 registered users:

```
!set max_users 0
!set max_extra_regs 1000
```

5.2.12 Setting up a nationalistic Hub

Verlihub can also restrict users to be within certain country zones. To restrict to US and Canada use:

```
!set cc_zone1 :CA:US:
!set max_users 7000
!set max_users0 100
!set max_users1 = 6900
```

5.2.13 Multiple Language support

Some text of Verlihub can be translated, specifically those that hub sends to users. By default, these special fields are not in the database. To enable this, you need to:

```
!set save_lang 1
```

Restart the hub, then:

```
!set save_lang 0
```

You need to reset *save_lang* to 0, it is only needed once to create special settings in the database.

Now in the database you have (in the SetupList table) some variables with their english values (they have the format lang_en in the table). Whatever you change in these variables will be loaded next time. If you like you can keep the lang_en variables and make a copy to lang_whatever. If you do this, you need to tell Verlihub to use the new ones on next startup, by adding a line in dbconfig in the configuration folder (in this guide, the file will be /etc/verlihub/dbconfig). All you need to do is add:

```
lang_name = lang_whatever
```

Verlihub will then use the language specified when the hub is restarted.

5.2.14 Bending user limit rules

Verlihub allows you to "bend" the rules for certain people joining the hub. When the limit in max_users is reached, and either a Registered user, an OP, or an Admin OP joins, Verlihub checks these three values:

- max_extra_regs - Amount of registered users (class 1) allowed to join hub after the hub is full
- max_extra_vips - Amount of extra VIP users (class 2) allowed to join hub after the hub is full
- max_extra_ops - Amount of OPs (class 3) allowed to join hub after the hub is full
- max_extra_admins - Amount of Admin OPs (class 5) allowed to join hub after the hub is full

So if you would like to have no registered members join after the hub is full, but you still want 5 OPs and 5 Admin OPs allowed to join, use this set of commands:

```
!set max_extra_regs 0
!set max_extra_vips 5
!set max_extra_ops 5
!set max_extra_admins 5
```

5.2.15 Increasing file descriptor limits for connections >= 10mbit

If for example you have a 10mbit line, you need to increase the default value for the file descriptor limits to 4096. If however, you have a 100mbit line, you need to increase your file descriptor limits by a larger amount or you will be limited to how many users are in your hub (for example, increase it to 10240.)

To do this, you need to change the following files:

In **/etc/security/limits.conf**, add the lines:

```
*     soft    nofile  4096
*     hard    nofile  4096
```

In **/etc/pam.d/login**, add:

```
session required /lib/security/pam_limits.so
```

Then increase the system-wide file descriptor limit by adding the following lines to the **/etc/rc.d/rc.local** startup script:

```
echo 16256 > /proc/sys/fs/file-max
echo 24576 > /proc/sys/fs/inode-max
```

Also, you can run these commands at the command line and/or you can add them to your .bash_profile so you don't need to re-enter them.

You will then need to tell the system to use the new limits (bash):

```
ulimit -n unlimited
```

For csh/tcsh:

```
unlimit descriptors
```

Verify this has raised the limit by checking the output of `ulimit -a` (bash) or `limit` (csh, tcsh).

Then you can start your hub.

5.3 Configurable variables

The table below lists all of variables that are configurable in verlihub (this list is more than likely incomplete). To change their values, use the !set command when logged into the hub as a Master user.

General Hub settings	
Variable	**Explanation**
hub_name	The name of your hub.
hub_desc	A description of your hub.
hub_host	What people will use to connect to your hub. (An example: myhub.no-ip.com)
hub_owner	Owner that runs the hub.
hub_security	This is the name of the Hub Security bot.
hub_security_desc	This is the description of the Hub Security bot.
opchat_name	This is the name of the OpChat bot.
opchat_desc	This is the description of the OpChat bot.
show_tags	Show tags to users? 0=hide from all, 1=show tags, 2=show tags to OPs only. Default=2.
Hub Connection settings	
Variable	**Explanation**
listen_port	The port the hub listens on. Other hubs use port 411, but you need to run verlihub as root to do that. Default=4111.
extra_listen_ports	Extra ports the hub listens on.
listen_ip	If you have more than one network card in your PC, enter the IP address to listen on. If you don't have more than one network card, no change is necessary.
dns_lookup	Reverse DNS lookup when user joins. Needed to see hostmask of user. If you have a slow DNS server this can slow down the hub.
max_users	Maximum number of users allowed in your hub.
max_upload_kbps	Maximum upload speed hub is allowed to use. This is VERY useful for protecting the hub against lag in larger hubs. Setting the right limit will prevent the hub from being to accept more then it can handle at any given time thus preventing a backlog of actions.
min_frequency	Somewhat automatic antilag system, higher values will allow lower lag; 0 means disabled. Largest usable value is 3. Normal values would be between 0.1 and 1.5. If you want to try to break records, set to 0 (though it's not recommended.) Keep in mind that the lower this value is, the less extra resources the hub has to help deal with any backlog of actions. Recommend you leave this as default.

min_class_use_hub	Setting this variable to a number of class will prevent a user from being able to download if their class is below the set limit. This will over-ride the next variable below if not 0
min_class_use_hub_passive	The same as above only this one is specifically for passive users.
Nickname Rules	
Variable	**Explanation**
min_nick	Minimum nickname length. Default of 3.
max_nick	Maximum nickname length.
nick_chars	Allowed characters in username.
nick_prefix	To force a connecting user to have a prefix before their nick, example: [GOD] will tell all connecting users to set their nick as [GOD]nickname
nick_prefix_cc	Explained above but used when cc zones have been defined.
Share Rules	
Variable	**Explanation**
min_share	Minimum amount that people must share in your hub in megabytes. People that share less will not be allowed to join.
min_share_reg	If you want your registered users to have a different minimum share level, use this.
min_share_vip	If you want your VIP users to have a different minimum share level, use this.
min_share_ops	If you want your OPs to have a different minimum share level, use this.
min_share_factor_passive	Multiplication factor for the minimum share of passive users (a setting of 2.0 means passive have to share twice the min_share setting)
min_share_use_hub	If this is higher then min_share then users that have less then this will not be able to search and download, but will be allowed to join the hub
max_share	If you want to prevent users from sharing too much, enter the max allowed to share in MB.
max_share_reg	If you want to prevent registered users from sharing too much, enter the max allowed to share in MB.
Ban settings	
Variable	**Explanation**
tban_kick	Length of time people must wait after being kicked to rejoin. In seconds. Default=300 seconds.
tban_max	Maximum length of time an OP can ban someone for. In seconds.

hide_all_kicks	Hides all kick messages from users.
Login settings	
Variable	**Explanation**
int_login	Number of seconds user has to wait after disconnection before being allowed to reconnect.
send_user_info	When people join, the hub will send them information about themselves. Set to 0 (zero) to disable, or 1 to enable. Default=On.
send_user_ip	When people join, the hub will send them their IP address. Set to 0 (zero) to disable, or 1 to enable. Default=On.
nicklist_on_login	Show nicklist when users log in.
allways_ask_password	Always ask users for a password, whether they are registered or not.
chat_default_on	Enable the chat window by default. 1=yes, 0=no. If no, users need to use !chaton to get messages.
Custom Hub Messages	
Variable	**Explanation**
msg_chat_onoff	
msg_change_pwd	You can customize your user's change password message by putting a message in this variable.
Custom Error Messages	
Variable	**Explanation**
msg_banned	You can customize the error message that indicates the user is banned by putting a message in this variable.
msg_hub_full	You can customize the error message that indicates the hub is full by putting a message in this variable.
msg_nick_prefix	
msg_downgrade	You can customize the error message that indicates the client is too new by putting a message in this variable.
msg_upgrade	You can customize the error message that indicates the client is outdated by putting a message in this variable.
msg_replace_ban	
Custom Welcome Messages	
Variable	**Explanation**
msg_welcome_guest	If you want to welcome your Class 0 members, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.

msg_welcome_vip	If you want to welcome your registered members, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_vip	If you want to welcome your VIP members, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_op	If you want to welcome your OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_cheef	If you want to welcome your Cheef OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_admin	If you want to welcome your Admin OPs, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
msg_welcome_master	If you want to welcome your Master user, fill in a welcome message here. Use %[nick] to use the person's nickname in the message.
Maximum User Extensions	
Variable	**Explanation**
max_extra_regs	If the hub is full, allow this many more extra registered users to join.
max_extra_vips	If the hub is full, allow this many more extra vip users to join.
max_extra_ops	If the hub is full, allow this many more OPs to join.
max_extra_admins	If the hub is full, allow this many more Admin OPs to join.
Public Hub List settings	
Variable	**Explanation**
hublist_host	
hublist_port	
hublist_send_minshare	When set to 1, a minimum share field is added automatically to the hublist registration description.
timer_hublist_period	Number of seconds between two hublists registrations.
Hub redirection	
Variable	**Explanation**
redir_host_max	Set this to 0 (zero) and redir_host0 to your hub's DNS name to disable redirection.
redir_host0	Enter hosts to redirect to in case users aren't allowed in your hub, or leave them blank. Dont forget to update redir_host_max if u decide not to leave these blank.
redir_host1	
redir_host2	

redir_host3	
redir_host4	
redir_host5	
redir_host6	
redir_host7	
redir_host8	
redir_host9	
Search settings	
Variable	**Explanation**
int_search	Minimum number of seconds user has to wait before being allowed to search again.
int_search_reg	Minimum number of seconds registered users have to wait before being allowed to search again.
min_search_chars	Minimum number of characters allowed in a search. (This can be a useful means of controlling wildcard searches such as 'mp3' which for a passive user would return thousands upon thousands of search results back through the hub.
max_passive_sr	Maximum number of search responses a passive user is allowed to recieve. This is useful for lag and bandwidth control since passive users totally rely on the hub to return search results.
DC Tag settings	
Variable	**Explanation**
tag_allow_none	Allow clients to NOT report a tag? 1=yes, 0=no. Yes by default.
tag_sum_hubs	The count of numbers to sum and use as the HUBS count from the H:X/Y/Z part of tag
tag_min_class_ignore	Which lowest class users are not to be checked for tags
tag_max_hs_ratio	Enter the minimum hub/slot ratio here. Users that are below this are refused connections to the hub. For example, setting to 1 means user needs to have 1 slot open per hub.
tag_max_hubs	Set this to restrict how many hubs your user can be in concurrently.
tag_min_version_plusplus	Use the _min_version_ tags to specify a minimum version for the client allowed in the hub.
tag_min_version_dcgui	
tag_min_version_odc	

tag_min_version_dc	
tag_min_version_dcpro	
tag_min_version_strongdc	
tag_min_version_idc	
tag_min_version_zdc	
tag_max_version_plusplus	Use the _max_version_ tags to specify a maximum version for the client allowed in the hub.
tag_max_version_dcgui	
tag_max_version_odc	
tag_max_version_dc	
tag_max_version_dcpro	
tag_max_version_strongdc	
tag_max_version_idc	
tag_max_version_zdc	
tag_allow_sock5	We all know what a socks connection is, this can prevent a user connecting to your hub using such a method if set to 0
Country Code User Definitions	
Variable	**Explanation**
cc_zone1	Enter the countries for the first country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
cc_zone2	Enter the countries for the second country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
cc_zone3	Enter the countries for the third country code zone. The format is `:XX:YY:ZZ:`. It has to start and end with a colon, and all country codes are separated by colons. There can be any number of codes (example codes: CZ; UK; US; SE. **NOTE:** These need to be capitalized!)
max_users0	If a user joins that doesn't belong in the Country Code list specified below, this user limit applies.
max_users1	User limit that applies to cc_zone1
max_users2	User limit that applies to cc_zone2
max_users3	User limit that applies to cc_zone3

IP Address User Definitions	
Variable	**Explanation**
ip_zone4_min	If you want to restrict users by IP address, enter the minimum IP address for the first IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone4_max	If you want to restrict users by IP address, enter the maximum IP address for the first IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone5_min	If you want to restrict users by IP address, enter the minimum IP address for the second IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone5_max	If you want to restrict users by IP address, enter the maximum IP address for the second IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone6_min	If you want to restrict users by IP address, enter the minimum IP address for the third IP zone here. (Verlihub allows you to define 3 IP zones.)
ip_zone6_max	If you want to restrict users by IP address, enter the maximum IP address for the third IP zone here. (Verlihub allows you to define 3 IP zones.)
max_users0	If a user joins that doesn't belong in the IP range specified below, this user limit applies.
max_users4	User limit that applies to ip_zone4_*
max_users5	User limit that applies to ip_zone5_*
max_users6	User limit that applies to ip_zone6_*
Miscellaneous settings	
Variable	**Explanation**
cmd_start_user	The prefix for user commands. Default '+'.
cmd_start_op	Prefix for OP commands. Default '!'.
report_dns_lookup	This will report to opchat by default the results of a reverse dns lookup.
max_message_size	This will over-ride the earlier mentioned flood control settings if set correctly. Works in bytes i.e. 1024 bytes of data allowed in a single message default is 10240 allowing a huge message. This applies to ALL classes including master (class 10)

int_chat_ms	This is the time period between each message a user can send to mainchat. A value of 1000 would make a user wait 1 second at least before he can send another message to mainchat. Can be used for flood control.
show_email	Set this to 0 if you wish to optimise your nicklist by removing all email fields. Set to -1 to disable and show all email addresses that user specify in their client settings.
show_speed	See show_email.
show_share	See show_email.
show_desc_len	The same as the show_email variable except with this you can specify a number of characters that are allowed. i.e. 10 would allow a user description length of 10 characters. -1 to disable and show entire description.
int_nicklist	A value of 30 by default meaning a user can only request a nicklist update once every 30 secs, useful for lag and bandwidth control.
desc_insert_mode	Insert 'A', 'P', or '5' into user's description ? (1 = yes/0 = no) A meaning Active P meaning passive and 5 meaning a socks connection type.
use_reglist_cache	Set to default by one Verlihub will cache the entire reglist for query upon entry to the hub. If this is turned on then any external apps used for adding users directly into the database will not be known in the reglist cache until either !reload is typed in the hub or the hub reloads its own config variables, a time setting which can be manipulated.
use_penlist_cache	The exact same as the last variable but this one deals with the banlist and kicklist and temp rights.
save_lang	Set to 1 to enable multi-language. See this guide for more details and rememeber to turn it off after you have saved the lang as described in the guide earlier.
timer_conn_period	It's probably best that you do not mess with this variable as it does not really effect performance in the way people seem to think it does.
timer_serv_period	See previous variable.
timer_reloadcfg_period	This is a time setting in seconds that tells verlihub to reload its settings and reglist and banlist cache from the database. Default is 300 seconds (5 minutes) meaning every 5 minutes verlihub will reload the database.
delayed_search	Set to 1 by default this will impose an initial delay specified by int_search on a connecting user. Useful for search bots that search spam hubs.
delayed_myinfo	The same as above, See int_myinfo for a better description.

step_delay	Do not mess with this setting unless you know what you are doing.
bc_reply	When you set to disable (disable), then the !bc commands don't send you away replies from all back to your PM
log_level	5 log level settings 0 being disabled 1 being slight 2 3 4 5 etc etc where 5 logs everything it can log. Use with caution and only when debugging as this feature can significantly impact resources.
check_ctm	By default this is set to 1 which will force the hub to varify a connecting active users connection (make sure he is who he claims to be) 0 to disable this check (WARNING: setting to 0 means people can use your hub to generate what is known as a ctm flood, to attack other hubs or even your own). There used to be 3 levels for this, there is no longer, its either on or off theres no inbetween. When on if a user cannot be varified he/she will be disconnected from the hub.
check_rctm	The same as above but for passive users.
check_asearch	Performs the same as check_ctm only this one will kick in when a user searches, meaing if check_ctm is disabled a users ctm will be checked when he attempts a search. See check_ctm for better description.
optimize_userlist	Set this to 1 if you wish to upload the userlist more slowlier to users upon connecting. How slowly is determined by the next variable below.
ul_portion	See optimize_userlist.
int_flood_pm_period	A means of flood control only this one applies to private messages. This is a time setting in seconds and will measure how many private messages you are trying to send a user over a given time period specified here.
int_flood_pm_limit	This setting is a numerical value. Example: value of 2 will disconnect a user if they try to spam another more then twice in quick succession.
timeout_myinfo	Genrally you should leave this at its default setting. Its self explanitory.
hub_version	This is a hard value determined by the source code. If you change it then upon next restart the hub will change it back again.
hub_version_special	You can specify an extra version string here that will be appended to the hard coded version string of the hub.
classdif_reg	This value is somewhat confusing for many. Lets explain it with an example. When set to the value of 2 a user can only register another user that is maximum 2 classes below him/her.
classdif_kick	See above.
classdif_download	See above.
classdif_pm	See above.

min_class_register	This is self explanitory. If you set this to 3 then it will over-ride any other settings and ONLY OPS or higher will be able to register a new user.
min_class_bc	This setting is again self explanitory, a setting of 5 means only admins can send broadcast messages.
min_class_bc_regs	See above. The regs extension is referring to registered users. This over-rides the value above when set.
min_class_redir	Self explanitory. Applies to manually redirect users from your hub.
dest_report_chat	This setting is the output of the +report command. By default all +reports will be shown in opchat.
dest_regme_chat	See above. Applies to the +regme command.
dest_drop_chat	See above. Applies to users who are dropped from the hub by other users i.e. OP's who use the !drop command,
default_password_encryption	The ecryption method which will be used (if available) to encrypt passwords that will be set. Effective immediately. (1 - ENCRYPT (default), 0 - not encrypted - plain text, 2 - MD5). If given method is not available, another encryption method is used or at worst the plain text password is stored.
timeout_key	timeout (in seconds) from the connection until the first response
timeout_nick	timeout (seconds) from the key response, until the reception of nick
timeout_myinfo	timeout (seconds) - from the nick until the myinfo (description, share, tags,etc..)
timeout_flush	timeout (seconds) (reserved, not used)
timeout_login	timeout (seconds) from the connection until the appearance in the userlist
timeout_setpass	timeout (seconds) from login until user sets his/her initial password
max_class_int_login	maximal class that get int_login seconds long tempban on every login (this is added mainly for debugging purpose)
mainchat_class	The minimum class that can use the mainchat.
topic_mod_class	The minimum class that can set the hub topic.
plugin_mod_class	The minimum class that can issue plugin commands.
trigger_mod_class	The minimum class that can issue trigger moderation commands.
disable_regme_cmd	Set this to 1 to prevent people using the +regme command.
disable_me_cmd	See above.

disable_usr_cmds	Set this to 1 to temporarily disable all + commands for users below class 1.
max_class_int_login	maximal class that get int_login seconds long tempban on every login (this is added mainly for debugging purpose)

6. Managing your hub

Verlihub has various commands that range from adding registered users to banning them from the hub. This section has been broken into 5 parts: User Management, Kicking users, Banning users, Sending messages to users, and Muting users.

6.1 User management

This section of the manual deals specifically with commands and instructions for user management. Before you do any user management, make sure you are familiar with the next section, User classes.

6.1.1 User classes

There are several "classes" of users that can exist in verlihub. Below is a summary of the class types:

Class	Tag	Description
0	Regular users	Any user that connects to the hub
1	Registered users	Password protected nickname
2	VIP users	Special share limit
3	Operator user	Special share limit, kick/ban users, can register level 1 users
4	Cheef user	Mainly is allowed to register users and VIP members, but not allowed to create OPs
5	Admin user	Can create OPs, broadcast messages and change hub settings via !set
10	Master user	This user can do everything, including create more admin users, can't create another master user

6.1.2 Adding a registered user

First some clarification on who can add users. Normal users, VIP users, and Normal OPs (Level 3) can not add a registered user. Cheef OPs (Level 4) can add Registered users, and Admin OPs (Level 5) can create VIP users and Registered users. Master users (Level 10) can NOT create another Master user, however they can create Admin OPs, Cheef OPs, OPs, VIP users, and Registered users. All of this is summarized in this table below:

User Level	Can create users of level
0 (Normal users)	can't add users
1 (Registered user)	can't add users unless min class register is set for them
2 (VIP user)	see above
3 (Regular OP)	see above
4 (Cheef user)	By default can add classes 1 and 2
5 (Admin user)	By default can add 1 2 and 3
10 (Master user)	Can add all regardless of any settings

With that out of the way, to register a user use the command **!regnewuser <nick> <class>**. As an example, here's how to create the user 'ARegUser' as a normal registered user:

```
!regnewuser ARegUser 1
```

NOTE: The user will need to use the +passwd command to register a password! They will be kicked from the hub if they don't change the password in 5 minutes or the time set by you in the variables settings!

TIP: If you have some users that can't figure out how to change their password you can do it for them by using the command **!regpass <nick> <password>**.

6.1.3 Removing a registered user

Only Admin OPs (Class 5) are allowed to remove registered users. Removing a registered user in done using the **!regdelete <nick>** command. For example, to remove user 'BadUser' use:

```
!regdelete BadUser
```

6.1.4 Temporarily disabling an account

Verlihub allows you to temporarily disable a registered nickname without permanently removing it from the database. This is done using the **!regdisable <nick>** command. **!regenable <nick>**. This is example of disabling user 'DisableMe' and then enabling it right away:

```
!regdisable DisableMe
!regenable DisableMe
```

6.1.5 Changing a user class

Ah yes, promotions (or demotions?) always seem to happen depending on user's behaviour. Should this ever happen, you can simply change the user's class with the command **!regclass <nick> <newclass>** command. The rules for adding new users apply here (see here). For example to change the user 'NewOP' to class 3 (Normal OP) use:

```
!regclass NewOP 3
```

To temporarily OP someone, use the command **!class <nick> <newclass>**. So to temporarily OP the user 'TempOP' use:

```
!class TempOP 3
```

6.1.6 Getting information on a user

There are two ways to get information about a user in Verlihub, both with slightly different information. If you want to see the user's registration information, use **!reginfo <nick>** and it will show you the information related to the user's registration. For example, to see information on user 'Someuser':

```
!reginfo Someuser
```

The other information will show simply the IP address, and the Hostname (if dns_lookup = 1 in the configuration). This command is simply **!getinfo <nick>**. So, to get information on 'Someuser' again, use:

```
!getinfo Someuser
```

If you want to get a list of nicknames by providing the IP address, use the command **!whoip <IP address>**. There are also two other ways to get information by IP:

- **!whorange <lower IP address>..<upper IP address>** (shows nicknames for a range of addresses)
- **!whorange <ip address>/<left mask bits>** (shows nicknames by subnet)

6.1.7 Registered user kick/ban protection

If you want to protect a registered user from accidental kicks, you can do this with the command **! regprotect <nick> <protect_class>**. What this does is allow you to grant protection from being kicked from certain classes of users. For example, to protect 'VIPUser' from normal OPs (Class 3) use this command:

```
!regprotect VIPUser 3
```

and this will prevent user from being kicked by any Class 3 user.

6.1.8 Recording a note about a user

To add a note in the registration database, use the **!regset <nick> note_op <msg>** command. An example:

```
!regset NewRegUser note_op <This is my friend, that's why I regged him>
```

6.1.9 Changing passwords

Once a user has registered a password once, he/she will need to inform the OP if for whatever reason they want to change it. The user will not be allowed to change the password without an OP knowing. To allow the user to set the password use the command **!regpass <nick>**. So, to allow user 'NeedPWChanged' to have their password changed:

```
!regpass NeedPWChanged
```

After this has been done, the user will be prompted to change their password. They will be kicked from the hub if they don't change the password in 5 minutes.

TIP: If you have some users that can't figure out how to change their password you can do it for them by using the command **!regpass <nick> <password>**.

6.1.10 Muting a user

You can also "mute" a given user temporarily from chatting. Using command **!gag <nick> [<time>]** will keep the user quiet for the time specified (the default is 2 days) or !ungag <nick> is used. For example to mute user 'Annoying':

```
!gag Annoying
```

And to unmute the user:

```
!ungag Annoying
```

6.1.11 Preventing a user from searching the hub

You can prevent users from searching the hub if they are abusing it. Using command **!nosearch <nick> [<time>]** will prevent the user from searching for the time specified (default is 2 days) or !unnosearch <nick> is used. For example to prevent user 'Annoying' from searching:

```
!nosearch Annoying
```

And to allow searches again:

```
!unnosearch Annoying
```

6.1.12 Preventing a user from private messaging others

You can prevent users from messaging others if they are abusing it. Using command **!nopm <nick> [<time>]** will prevent the user from sending PMs for the time specified (default is 2 days) or !unnopm <nick> is used. For example to prevent user 'Annoying' from sending messages:

```
!nopm Annoying
```

And to allow messages:

```
!unnopm Annoying
```

6.1.13 Preventing a user from connecting to others

You can prevent a user from connecting to others in your hub. Using command **!noctm <nick> [<time>]** will not allow the nickname specified to connect to anyone in the hub for the time specified (default is 2 days). For example to prevent user 'Annoying' from connecting to anyone:

```
!noctm Annoying
```

And to allow messages:

```
!unnoctm Annoying
```

6.1.14 Manipulating the registered users table directly

Verlihub allows you to manipulate the registered users table directly, using the **!regset** command. The format for this command is:

```
!regset <nick> <variable> <value>
```

Some of this information is displayed to the user when the login or use the *+myinfo* command. Here are the items you can change in the registered users table:

Variable	Description
nick	The user's nickname.
class	The class the user belongs to.
class_protect	User is protected from being kicked by this class and lower.
class_hidekick	User won't see kicks from OPs this class or lower.
hide_kick	Flag to indicate whether User sees any kick messages.
hide_share	Flag to indicate whether Users share should be shown or not.
reg_date	Date the nickname was registered. (UNIX timestamp!) **This HAS to be set!**
reg_op	The OP that registered the nickname.
pwd_change	Flag to indicate if user needs to change his/her password.

pwd_crypt	Type of encryption used on the password.
login_pwd	The user's password.
login_last	Last time User logged in. (UNIX timestamp!)
logout_last	Last time User logged out. (UNIX timestamp!)
login_cnt	How many times user has logged into the hub.
login_ip	IP address of user when they last entered hub.
error_last	Last error message User reported. (UNIX timestamp!)
error_cnt	Total number of errors User has had.
error_ip	IP address of User when last error occured.
enabled	Flag to indicate whether account is enabled or not.
email	User's email address
note_op	A field you can use to leave yourself a note regarding this user.
note_usr	Another note field.
msg_welcome	Custom personal welcome message.

For example, to change the name of the person who registered user Test use:

```
!regset Test reg_op TheHub
```

6.2 Kicking users

Kicking users temporarily removes them from the channel. They will not be able to rejoin immediately, depending on the settings of *tban_kick* in the Hub settings. (The default is 5 minutes.) Kicking someone is done using the **!kick <nick> <reason>** command. You have to specify a reason to kick the user from the hub. For example to kick 'BadUser' because he isn't sharing enough use:

```
!kick BadUser You are not sharing enough.
```

If you don't want a public kick message to show in the chat window, use **!hidekick**. Alternatively, set the configuration variable *hide_all_kicks* to 1 to hide all kick message.

If you want to ban the IP of a user you just kicked and can only remember the nickname, use **!banip this nick <nick> <reason>**.

6.2.1 Banning during a kick

If you want to kick and ban a user with one command, simply add '_BAN_' somewhere in the kick reason. For example, to kick and ban 'BadUser' for not sharing enough:

```
!kick BadUser You are not sharing enough. _BAN_
```

Also, you can temporarily ban a user by appending the time unit after _BAN_ (for example, a 2 day ban you would use _BAN_2d). See Temporarily banning a user for more information.
This method bans both IP address and nickname.

6.3 Banning users

There are various ways to ban users fromt the hub. They are outlined below. **The best way to ban is during a kick. See Banning during a kick for information.**

6.3.1 Temporarily banning a user

If you need to, you can temporarily ban a user. First of all, you will need to know the time units that are available to you. They are illustrated in this table:

Time Unit	Abbreviation
Second	s
Minute	m
Hour	h
Day	d
Week	w
Month	M
Year	y

With that out of the way, just append the time unit after any of the ban commands, with an underscore seperating them. For example, to temporarily ban someone for 2 weeks, use:

```
!ban_2w BadPerson You have been temporarily banned.
```

6.3.2 Adding a ban (IP and nickname)

See Banning during a kick for information on how to do this.

6.3.3 Adding an IP ban

There are two commands to ban an IP address. The first is **!ban <ip> <reason>**, and the second is **!banip <ip> <reason>**. These commands are the same, it does not matter which one you use.

6.3.4 Adding a nickname ban

To ban a nickname, use the command **!bannick <nick> <reason>**.

If you want to ban the IP of a user you just kicked and can only remember the nickname, use **!banip this nick <nick> <reason>**.

6.3.5 Adding a hostname ban

Note: You need to configure variable *dns_lookup* for this to work. If you have a slow DNS server this can slow down the hub.

All of the examples will use hostname someuglynumbers.provider.whatever.com as an example.

Ban Command	Explanation
!banhost2 <hostname> <reason>	This will ban the second level of the given (in the example, whatever.com would be banned.)
!banhost3 <hostname> <reason>	This will ban the third level of the given (in the example, provider. whatever.com would be banned.)
!banhostr1 <hostname> <reason>	Bans the leftmost part of the hostname (in the example, "someuglynumbers" would be banned.)

6.3.6 Adding an email ban

To ban an email address, use the command **!banemail <email> <reason>**.

6.3.7 Adding a nick prefix ban

To ban a nickname prefix, use the command **!banprefix <nick_prefix> <reason>**.

NOTE: This one will affect the performance of users joining the hub, the more prefixes you ban the slower it will be!

6.3.8 Adding a share size ban

To ban a share size, use the command **!banshare <exact_share_size_in_bytes> <reason>**.

6.3.9 Banning a range of IP addresses

There are two ways to ban a range of addresses. The first way is simply to specify a range of IP addresses to ban using **!banrange <ipmin>..<ipmax> <reason>**.

The second way is to ban by subnet. To do this, specify an IP and the subnet mask to ban, using the command **!banrange <ip>/<left_mask_bits> <reason>**

NOTE: This may not work on big-endian machines!

6.3.10 Getting information on bans

If you want to get information about a ban, there is the **!infoban <nick_or_ip_or_whatever>** command. This method will try to list all possible matches for what is entered.

If you want to specify a specific type of ban use these three commands:

- **!infobannnick <nick>**
- **!infobanip <ip>**
- **!infobanrange <any_ip_from_the_range>**

6.3.11 Removing bans

You have to provide a reason for the unban. Even a dot (".") is allowed for the reason. The unbans are stored in database, and the Master user can view them.

This is a list of the various unban commands:

Unban Command	Explanation
!unban <ip_or_nick> <unban_reason>	Standard method for removing bans.
!unbanip <ip> <unban_reason>	Will remove only existing IP bans.
!unbannick <nick> <unban_reason>	Removes nickname bans.
!unbanhost2 <hostname> <unban_reason>	Removes a second-level hostname ban.
!unbanhost3 <hostname> <unban_reason>	Removes a third-level hostname ban.
!unbanhostr1 <hostname> <unban_reason>	Removes the leftmost part of a hostname ban.
!unbanemail <email> <unban_reason>	Removes an email ban.
!unbanprefix <prefix> <unban_reason>	Removes a nickname prefix ban.
!unbanshare <exact_share_in_bytes> <unban_reason>	Removes a share size ban.
!unbanrange <low IP value>..<high IP value> <unban_reason>	Removes a range of IP bans.

!unban _nickban_ <reason>	Removes ALL current nickbans
!unban _ipban_ <reason>	Removes ALL current ipbans
!unban _shareban_ <reason>	Removes ALL current sharebans
!unban _rangeban_ <reason>	Removes ALL current rangebans
!unban _tempban_ <reason>	Removes ALL current tempbans
!unban _emailban_ <reason>	Removes ALL current emailbans
!unban _hostban_ <reason>	Removes ALL current hostbans

6.4 Sending messages to multiple users

Verlihub provides three methods for sending messages to multiple users: a simple broadcast to everyone, a message to OPs only, and a message to all registered users.

6.4.1 Sending a message to everyone

If you ever need to send a message to all users connected to your hub, use the **!broadcast <message>** command. For example, to send a message to everyone that the hub is going down for maintenance:

```
!broadcast This hub is going down for maintenance and will be back in
2 minutes.
```

6.4.2 Sending a message to OPs only

There are two ways to send a message to OPs only. First of all, if there is an OPChat bot in your hub, simply send it a message and it will relay that message to all of the other OPs.

Secondly, use the **!ops <message>** command. For example:

```
!ops We don't have an OPChat bot!
```

6.4.2 Sending a message to registered users only

If you want to send a message to your registered users, use the **!regs <message>** command. For example, to tell the registered users they need to change their password:

```
!regs You need to reset your password. PM an OP for more details.
```

7. Plugins

Verlihub supports the use of plugins. The actual plugin development is beyond the scope of this guide. If

you are wondering how to compile a plugin, refer to the documentation that came with the plugin. The commands are listed here in case you decide to install one. **If you upgrade Verlihub, IT IS VITAL you recompile ALL of the plugins you have loaded!**

7.1 Direct Plugin support

Verlihub has a basic interface to load/unload plugins. See sections 7.1.1 - 7.1.4 for the commands.

7.1.1 Listing loaded plugins

To list all loaded plugins use:

```
!pluglist all
```

7.1.2 Loading a plugin

After you have downloaded and compiled a plugin, use this command to load it:

```
!plugin <plugin_filename.so>
```

7.1.3 Unloading a plugin

To unload a plugin use:

```
!plugout <plugin_name>
```

7.1.4 Reloading a plugin

To reload a plugin, use:

```
!plugreload <plugin_name>
```

7.2 The Plugin Manager plugin

There is a plugin manager available for Verlihub. It is available as a plugin (ironically) and allows you to list available plugins and turn them on and off. This is the preferred way to manipulate the plugins.

The Plugin Manager should have been compiled with Verlihub when you installed it; it is just not loaded automatically. The plugins are by default installed into /usr/local/lib. To load the Plugin Manager manually use this command:

```
!plugin libplug_pi.so
```

This will load it once only; when you restart you will need to load it again. Alternatively, to load the

plugin manager when Verlihub starts, create a symlink to your configuration directory (in this guide **/etc/verlihub** by issuing the following commands:

```
mkdir /etc/verlihub/plugins
ln -s /usr/local/lib/libplug_pi.so /etc/verlihub/plugins/libplug_pi.so
```

Now restart Verlihub, and the Plugin Manager should be loaded automatically. Gentoo users can use:

```
/etc/init.d/verlihub restart
```

to restart Verlihub.

To see if the Plugin Manager is working use this command to list loaded plugins:

```
!pluglist all
```

The hub should report **plugman x.x** if it is loaded.

7.2.1 Plugin Manager commands

The Plugin Manager commands have been put in this table for quick reference:

Command	Parameters	Description
!lstplug		List the registered plugins
!addplug	<nick[8]> [-p <"path"> [-d "desc"] -a <autoload (0/1)>	Registers a new plugin. *nick* is a short plugin nickname (8 characters); *path* is a relative or absolute path to the plugin's binary; *desc* is an optional description; *autoload* set to 1 to have it start when verlihub loads.
!delplug	<nick>	Unregisters the plugin.
!onplug	<nick>	Turns the plugin on.
!offplug	<nick>	Turns the plugin off.
!replug	<nick>	Reloads the plugin, turning it off then on.

7.3 Available plugins

There are various plugins available for Verlihub; however, it is better to check the WiKi as that document will always be more up to date. Also, the Plugin Manager may list the available plugins once it is loaded. A short list is provided here (remember, check the WiKi for more current plugins!):

The Plugin Manager	
Plugin Name	**Description**

plugman	This is the plugin manager for Verlihub, and is part of the verlihub download tarball.
Scripting Languages	
Plugin Name	**Description**
lua	This allows you to use simple scripts written with LUA.
perl	This allows you to use simple scripts written in Perl.
Developer Tools	
Plugin Name	**Description**
PluginScriptInterface	This is a special interface with better support of scripting languages.
Chat and Flood control	
Plugin Name	**Description**
forbid	This plugin allows you to filter main chat and private messages for forbidden words.
replacer	This plugin replaces given patterns in text. For example if someone types Windows in the main chat, you can have this plugin change it to Winblows.
funny	This one is amusing, it mixes up the chat messages. Install this one and don't tell your hub users. ;)
floodprot	This plugin offers more control from users flooding the hub.
diakritika	This plugin replaces some letters with other letters (to remove diatrics for example, or just to simply make all of the chat messages lowercase. This was included in the tarball of verlihub.
Miscellaneous	
Plugin Name	**Description**
stats	This periodically saves statistics in the database (i.e. users, share, upload, searching, etc...)
messanger	This plugin is used to send messages to offline users.
gagrange	This plugin is used to disable chat for all non registered users in a given IP range.
iplog	This logs IP addresses as users join and leave the hub.
chatroom	This will set up individual chatrooms for users. This would work similarly to OPChat. This is great if you have a multiple-language hub.
isp	This plugin allows you to check connection types, nickname prefixes, and minimum shares, among other things...

7.3.1 LUA Plugin

These commands are available after the LUA plugin is loaded:

Command	Description
!lualist	Lists loaded LUA scripts.
!luaload <script>	Loads specified LUA script.
!luaunload <script>	Unloads specified LUA script.

7.3.2 forbid Plugin

These commands are available after loading the forbid plugin:

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>!lstforbid</td><td>Lists forbidden patterns.</td></tr>
<tr><td>!addforbid <"pattern"> [-C <max_class>] [-f <flags>] [-r <"kick_reason">]</td><td>Adds a forbidden pattern.
Using the -C option, you can specify the classes to ignore filtering. Set to 1 to ignore registered users and up.
Using the -r option, you can specify a kick reason.
Using the -f option, you can specify which chats are screened and whether or not a message is sent to OpChat when a rule is broken. Using the table below, to screen main chat and private messages, use -f 3.
<table>
<tr><th>Value</th><th>Explanation</th></tr>
<tr><td>1</td><td>Screen public chat</td></tr>
<tr><td>2</td><td>Screen private messages</td></tr>
<tr><td>4</td><td>Notify OpChat when a rule is broken</td></tr>
</table>
</td></tr>
<tr><td>!modforbid <"pattern"></td><td>Modifies forbidden pattern. See parameters for adding.</td></tr>
<tr><td>!delforbid <"pattern"></td><td>Deletes forbidden pattern.</td></tr>
</table>

7.3.3 iplog Plugin

These commands are available after loading the iplog plugin:

Command	Description
!lastip <nick> [<limit=10>]	Shows last IP addresses for nickname specified.
!lastnick <ip> [<limit=10>]	Shows the last nicknames used by the IP address.
!iplog <ip>	Shows log history for ip address.
!nicklog <nick>	Show log history for nickname.

7.3.4 Replacer Plugin

These commands are available after loading the replacer plugin:

Command	Description
!getreplacer	Lists replaced words.
!addreplacer <word> <replacement word> <level to apply>	Adds a word replacement. If you specify level 1, registered users and up are immune to the word replacement.
!delreplacer <word>	Removes an entry.

7.3.5 Messanger Plugin

These commands are available after loading the messanger plugin:

Command	Description
+msgsend <nick> <topic/subject> <newline> <message text>	Sends a message to *nick* and will be displayed next time they log in. It is a good idea to send this to a registered nickname. Enter a *newline* character between the subject and the message! Some clients require ctrl+enter to do this!

7.3.6 Chatroom Plugin

There are two sets of commands for the chatroom plugin; a command set to manipulate the chatrooms and a set of commands you can use in the chatroom to invite users and such.

The commands to manipulate the chatrooms are:

Command	Description
!lstroom	Lists all chatrooms available.
!addroom <nick> <topic>	Create a chatroom named *nick* and sets the topic specified. When specifying the nickname, make sure you obey the nickname rules!
!delroom <nick>	Removes specified chat room.

The commands to manage the users in the chatrooms are:

Command	Description
+invite <online_users_nick> <invite message>	Invites a person into the chatroom with an optional invitation message.

+leave	Leaves chatroom; no messages from the chatroom will be received.
+out <nick>	Excludes a user from the chatroom. This is useful if you have users in the chatroom that normally do not have the rights to access the room.

7.3.7 Gagrange Plugin

These commands are available after loading the gagrange plugin:

Command	Description
!lstgagrange	Shows ip ranges that have been gagged.
!addgagrange <ip>/<left_mask_bits>	Adds IP range to gag list. Example: 1.2.3.4/24
!delgagrange <ip>/<left_mask_bits>	Removes an IP range from the gag list.

7.4 Getting plugins

Most, if not all of the plugins are currently in CVS, which requires CVS software installed. Google around to find the appropriate cvs package for your distribution.

Gentoo users can use the following command:

```
emerge cvs
```

After this is complete, browse the Verlihub CVS repository for plugins available to download. **NOTE:** Not all of the folders are plugins!!

To retrieve a plugin, use the following command:

```
cd ~
cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/verlihub login
<hit enter when asked for password>
cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/verlihub co <plugin_name>
```

After getting the source for the plugins, you need to build them using these commands:

```
cd ~/<plugin_name>
./configure
make
make install
```

Once the plugin is built, check the location of the modules (default: **/usr/local/lib**) to make sure they exist there. After this is done, use the Verlihub Plugin Manager to register them (if needed - do a **! lstplug** first), then you can turn them on and off. If they still do not load, remove them with the Plugin

Manager and re-add them (see here for more details on the Plugin Manager.)

7.4.1 LUA Plugin

The LUA plugin requires some special setup in order for it to work. The lua package needs to be installed from the LUA website.

Gentoo users can use this command to install it:

```
emerge lua
```

After LUA is installed, follow the instructions in the Getting Plugins section of the manual.

8. Hub command list index

This command list is more than likely incomplete, but should give you an idea of what verlihub can do.

Master User Commands (Level 10)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
10	!quit			Stops the hub.
Admin User Commands (Level 5)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
5	!getconfig	!gc		Prints a list of all configuration variables that can be inserted into the database.
5	!set	!=	<variable> <newvalue>	Sets verlihub configuration variable *variable* to *newvalue*.
5	!userlimit	!ul	<limit> [time - minutes]	Temporarily changes the user limit to *limit*. If *time* is not specified, 60 minutes is assumed.
5	!hublist			Sends your hub information to the public hublisting.
5	!reload			Reloads verlihub's configuration. **NOTE:** Some configuration changes require the hub to be restarted!

5	!regset	!r	<nick> <variable> <value>	Manipulates registered users table directly. See Manipulating the registered users table directly for more information.
5	!plugin		<plugin_name>	Loads specified plugin.
5	!plugout		<plugin_name>	Unloads specified plugin.
5	!pluglist		all	Shows loaded plugins.
5	!plugreload		<plugin_name>	Reloads specified plugin.
5	!protoall_hubname		*unknown*	Sends new hub name to all users.
5	!protoall_hello		*unknown*	Simulates a login.
5	!protoall_quit		*unknown*	Simulates a logout.
5	!protoall_pm		*unknown*	Sends a PM, similar to the ! broadcast command.
5	!protoall_chat		*unknown*	Sends a chat message, similar to if you typed it in the main chat window.
5	!protoall_redir		*unknown*	Forces hub redirection.
5	!protoall_any		*unknown*	You can try anything here, the pipe is added to the end automatically.
5	!protoactive_*		*unknown*	Sends the proto command to active users. Commands are similar to the !protoany commands above; for example, ! protouser_hubname.
5	!protohello_*		*unknown*	Sends the proto command to users that don't support the NoHello feature. Commands are similar to the !protoany commands above; for example, ! protouser_hubname.
5	!protouser_*		*unknown*	Sends the proto command to users. Commands are similar to the !protoany commands above; for example, !protouser_hubname.
Cheef User Commands (Level 4)				
Level	**Command**	**Aliases**	**Parameters**	**Description**

4	!broadcast	!bc	<msg>	Sends *msg* to all users.
4	!regnewuser	!rn	<username> [class]	Registers new *username*. If you do not specify *class*, the user is registered with a class of 1. You can not register users with a class higher than your class - 2
4	!regpasswd		<nick>	Allows registered *nick* to change his password (he has to use command +passwd <pass>).
4	!regdisable		<nick>	Disable a user's registration, but he can be re-enabled later. You should use this if you only want to punish a user for example.
4	!regenable		<nick>	Enable a user's registration that was previously disabled with ! regdisable.
4	!regdelete		<nick>	Destroys a registered user by removing him from the database. If you later change your mind, you must register the user all over again. You should use this if you want to permanently remove a user.
4	!regclass		<nick> <class>	Changes a previously registered user's class to <class>. You cannot change a user to a class higher than yours - 2.
4	!class		<nick> <class>	Temporarily changes *nick*'s class to *class*. Stays effective until user leaves and rejoins hub.
4	!regprotect		<nick> <class>	Protects a registered user from being kicked/banned etc. by a user of a class lower than *class*. The user does not have to be in the hub.

4	!protect		<nick> <class>	Protects an unregistered user from being kicked/banned etc. by a user of a class lower than *class*. The user must be currently connected to the hub for this to take effect.
4	!drop		<nick>	Disconnects user from hub without giving him a reason and without banning him.
Op User Commands (Level 3)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
3	!hubinfo			Shows information about the hub.
3	!kick		<nick>nbsp;<reason>	Kick the user from command line and give a ban equal to the hub's specified tempban time. If you right-click in your client and kick, it does the same thing. If you include _ban_time in the reason, then the ban is extended to the time you specify. Valid entries for time are Ns, Nm, Nh, Nd, NM, Nw, Ny. The IP and the nick are both banned. This is the most reliable and easiest way to ban.
3	!hidekick		<nick>nbsp;<reason>	Kicks user *nick* with *reason*. It does this quietly; it doesn't announce it to the hub users.
3	!ban		<ip> <reason>	Bans *ip* from hub with *reason*.
3	!banip		<ip> <reason>	Bans *ip* from hub with *reason*.
3	!bannick		<nick> <reason>	Bans *nick* from hub with *reason*.
3	!banhost1		<host> <reason>	Bans first level of *host* from hub with reason. This usually isn't too useful; for example in my.web. host.com it would ban '.com'.
3	!banhost2		<host> <reason>	Bans second level of *host* from hub with reason. For example in my.web.host.com it would ban 'host.com'.

3	!banhost3		<host> <reason>	Bans second level of *host* from hub with reason. For example in my.web.host.com it would ban 'web.host.com'.
3	!banhostr1		<host> <reason>	Bans leftmost part in the hostname.
3	!banprefix		<prefix> <reason>	Bans a user prefix in the nickname.
3	!banshare		<shareamount> <reason>	Bans a share amount (in bytes).
3	!banemail		<email> <reason>	Bans an email address reported by clients.
3	!banrange		<ipmin>...<ipmax> <reason> <ip>/<bitmask> <reason>	Bans range of IPs. Examples: *! banrange 1.1.1.1...1.1.1.4* or *! banrange 1.1.1.1/24*.
3	!ban_time		<ip> <reason>	This bans the IP for the specified time. If a nick with the <ip> has been previously kicked, then that nick is banned too and any user with the <ip> is dropped. If no nick with the <ip> was previously kicked then no nick is banned and no user is dropped. If a nick with the <ip> is later kicked, then that nick is banned for the same time as the <ip>. Valid entries for time are Ns, Nm, Nh, Nd, NM, Nw, Ny. So '!ban_1M <ip> <reason>' would ban an ip for one month with the reason. The best use for ! bannick is if you want to prevent a certain ip (no matter the nick) from entering the hub. If you wish to ban a user who is connected to the hub you should use !kick instead.

3	!bannick_time		<nick> <reason>	This bans only the nick of the user for the specified time. Valid entries for time are Ns, Nm, Nh, Nd, Nw, NM, Ny. So '! bannick_10m nick reason' would ban a nick for 10 minutes with the reason. The best use for ! bannick is if you want to prevent a certain nick (no matter the ip) from entering the hub. If you wish to ban a user who is connected to the hub you should use !kick instead.
3	!unban		<ip> <reason>	Unban IP and nick for the user. If a nick is associated with an ip or an ip is associated with a nick, then both are removed. This also removes bans that were issued with !kick.
3	!unbannick		<nick> <reason>	Removes a nickname ban.
3	!unbanhost1		<host> <reason>	Removes a first level host ban.
3	!unbanhost2		<host> <reason>	Removes a 2nd level host ban.
3	!unbanhost3		<host> <reason>	Removes a 3rd level host ban.
3	!unbanhostr1		<host> <reason>	Removes a leftmost host ban.
3	!unbanprefix		<prefix> <reason>	Removes a nickname prefix ban.
3	!unbanshare		<share> <reason>	Removes a share ban.
3	!unbanemail		<email> <reason>	Removes an email ban.
3	!unbanrange		<any_ip_from_range> <reason>	Specify and IP in the range that is banned and the entire IP block will be unbanned.
3	!hideme		<class>	Set this and users below *class* won't be able to see you.
3	!ccbroadcast	!ccbc	<msg>	If you have country codes in your nickname, this will send a message to only those with specified country codes. For example: *!ccbc :GB:US: Hi those from Great Britian and the US!..*

3	!flood		<nick>	Floods user off hub. Beware!
3	!infobanip		<ip>	Shows ban information for *ip*.
3	!infobannick		<nick>	Shows ban information for *nick*.
3	!infobanrange		<any_ip_from_range>	Specify an IP from the IP range that is banned, and it will show the ip range that is banned.
3	!infoban		_ipban_	Shows current IP bans.
3	!infoban		_nickban_	Shows current nickname bans.
3	!infoban		_banrange_	Shows current IP Range bans.
3	!lsban		100	Shows the 100 most recent bans.
3	!getip		<nick>	Shows ip address for *nick*.
3	!gethost		<nick>	Shows hostname for *nick*. This requires dns_lookup in verlihub configuration to be set to 1.
3	!getinfo		<nick>	Shows information about *nick*.
3	!whoip		<ip>	Attempts to find nickname for *ip*.
3	!whorange		<ipmin>...<ipmax> <ip>/<bitmask>	Attempts to show nicknames for the range of IP addresses. Examples: *!whorange 1.1.1.1...1.1.1.4* and *!whorange 1.1.1.1/24*
3	!regs		<msg>	Sends *msg* to all registered users.
3	!nosearch		<nick>	Prevents *nick* from searching the hub.
3	!unnosearch		<nick>	Allows *nick* to search, if previously prevented with ! nosearch.
3	!gag		<nick> [time]	Prevents the user from using the chat function until he restarts his session or until [time] has expired. If you leave the [time] blank, then the time is set for one week. Valid entries for [time] are Ns, Nm, Nh, Nd, NM, Nw, Ny. So '!gag 1h would stop the user from chatting for 1 hour or until he reconnected.

3	!nopm		<nick>	Prevents *nick* from PM'ing others in the hub.
3	!noctm		<nick>	Prevents *nick* from connecting to others.
3	!unnoctm		<nick>	Allows *nick* to connect to others, when prevented with !noctm.
3	!unnopm		<nick>	Allows *nick* to PM other users of hub, if they were denied with !nopm.
3	!ungag		<nick>	Allows *nick* to chat in main channel, if they were silenced with !gag.
VIP User Commands (Level 2)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
No commands				
Registered User Commands (Level 1)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
No commands				
Normal User Commands (Level 0)				
Level	**Command**	**Aliases**	**Parameters**	**Description**
0	+rules			Shows hub rules.
0	+faq			Shows frequently asked questions.
0	+motd			Shows message of the day.
0	+help			Shows help.
0	+report		<msg>	Sends a message to an OP, through OPChat.
0	+regme		<msg>	Sends a message to an OP, through OPChat.
0	+myip			Shows user their IP address.
0	+myinfo			Shows user information about themself.
0	+me		<msg>	Replaces '+me' with their nick and sends message to channel.

9. Troubleshooting

At this point, this section will remain empty unless something with a real need deserves to go here. =P

9.1 Reporting Bugs

Please use the forum to report bugs.